

**CLAIMS**

1. A network adapter capable of receiving data from a network, said adapter arranged to receive data at at least a lowest and a highest data rate;  
5 said adapter comprising:

sampling means arranged to sample said data;

10 an equaliser arranged to receive, and equalise, said data samples, and said equaliser capable of being trained to equalise data, at at least, each of said lower and higher data rates; and

training means capable of training said equaliser to equalise data;

15 wherein;

said training means is initially arranged to train said equaliser to receive data at said lower rate allowing data to be decoded and if upon decoding said adapter determines that said equaliser has been trained to equalise data  
20 at the incorrect rate to retrain said equaliser to equalise data at the correct rate.

2. An adapter according to claim 1 including a data buffer arranged to receive sampled data from said sampling means.

25

3. An adapter according to claim 2 wherein the data buffer is arranged to store data at the highest frequency.

4. An adapter according to any one of the preceding claims wherein the adapter comprises a second buffer, or a training sequence store buffer, arranged to receive and store training data held within the received data.
- 5 5. An adapter according to claim 4 wherein the training sequence store buffer is arranged to hold the training sequence at least until it has been determined that the equaliser has been trained to receive data at the correct rate.
- 10 6. An adapter according to claim 4 or 5 wherein the training sequence store buffer receives data from the sampling means.
- 15 7. An adapter according to any one of claims 2 to 6 wherein the adapter comprises data down-sampling means arranged to output a selection of the data held in a buffer to the equaliser.
- 20 8. An adapter according to claim 7 wherein the down-sampling means is arranged to output a selection of the data therein, by predetermining a constant n, and outputting every n<sup>th</sup> data sample from the buffer.
9. A method of training an equaliser to equalise a data sequence, which data sequence may be at one of at least a highest and a lowest data rate, said method comprising;- 25 obtaining a number of data samples by sampling said data sequence; training said equaliser to receive said data sample at said lowest data rate; decoding a portion of said data sequence with the trained equaliser to ascertain the correct data rate; and

retraining said equaliser if the equaliser has been incorrectly trained.

10. A method according to claim 9 wherein the method comprises using  
5 training data within a preamble of the data sequence to train the equaliser.

11. A method according to claim 9 or 10 wherein the received data sequence is buffered.

10 12. A method according to claim 11 wherein data is held in the buffer at the highest data rate.

13. A method according to any one of claims 10 to 12 wherein training data is held in a second, training sequence store, buffer.

15 14. A method according to any one of claims 9 to 13 wherein the method comprises sampling the data sequence at the highest data rate and subsequently using only a portion of the samples should the data sequence be at another data rate.

20 15. A method according to claim 14 wherein the method uses only a portion of the data samples by determining a number n and using only every  $n^{\text{th}}$  data sample.

25 16. A method according to any one of claims 9 to 15 wherein the highest and lowest data rates are part of the series  $x^y$ , where x and y are integers.

17. A method according to claim 16 wherein x is two.

20

18. A method according to any one of claims 9 to 17 wherein the equaliser is trained for each data sequence received.
19. A method according to any one of claims 9 to 18 wherein the method 5 detects an end of sequence marker within the data sequence and once this is detected returns to a state of waiting to train the equaliser.
20. A method according to any one of claims 9 to 19 wherein the data sequence comprises a preamble, the end of which is received before inputs 10 to the, or each, buffer are enabled for received data samples.
21. A computer readable medium having stored therein instructions for causing a processing unit to execute the method of any of claims 9 to 20.
- 15 22. A computer program arranged to cause a data sequence to be received, which data sequence may be at one of at least a highest and a lowest data rate, said program:
- obtaining samples of said data sequence;
- 20 training an equaliser to equalise said samples;
- decoding a portion of said data sequence using said trained equaliser; and
- 25 re-training said equaliser if the equaliser has been trained incorrectly to the incorrect data rate.
23. An interface including an adapter according to any one of claims 1 to 8.

30

24. An interface according to claim 23 arranged to interface a computer or computer peripheral to a network.
  
25. An interface according to claim 23 or 24 provided as any of the following: a PCI card, an ISA card, a USB peripheral, a Firewire peripheral, a PCMCIA card, a MODEM riser card.